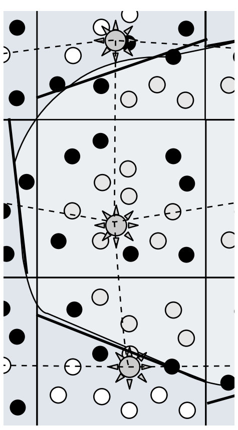
APPLY GEOMETRIC DUALITY TO ENERGY-EFFICIENT NON-LOCAL PHENOMENON AWARENESS USING SENSOR NETWORKS

JIE LIU AND FENG ZHAO, MICROSOFT RESEARCH
PATRICK CHEUNG, PALO ALTO RESEARCH CENTER
LEONIDAS GUIBAS, STANFORD UNIVERSITY



A wireless network of sensors can cover a large geographical region, and hence can be used to detect and track non-local phenomena which cannot be captured by any individual sensor.

ABSTRACT

A powerful concept to cope with resource limitations and information redundancy in wireless sensor networks is the use of collaboration groups to distill information within the network and suppress unnecessary activities. When the phenomena to be monitored have large geographical extents, it is not obvious how to define these collaboration groups. This article presents the application of geometric duality to form such groups for sensor selection and non-local phenomena tracking. Using a dual-space transformation, which maps a non-local phenomenon (e.g., the edge of a half-plane shadow) to a single point in the dual space and maps locations of distributed sensor nodes to a set of lines that partitions the dual space, one can turn off the majority of the sensors to achieve resource preservation without losing detection and tracking accuracy. Since the group so defined may consist of nodes that are far away in physical space, we propose a hierarchical architecture that uses a small number of computationally powerful nodes and a massive number of power constrained motes. By taking advantage of the continuity of physical phenomena and the duality principle, we can greatly reduce the power consumption in non-local phenomena tracking and extend the lifetime of the network.

INTRODUCTION

A wireless network of sensors can cover a large geographical region, and hence can be used to detect and track non-local phenomena that cannot be captured by any individual sensor. Because of its dense spatial sampling and multimodality sensing, the network can assemble information from spatially diverse sources to improve the signal-to-noise ratio. The redundancy in the network can ensure a certain degree of robustness against node failures. The network may be quickly deployed for a particular application, and the ubiquity and low-cost nature of the micro-electromechanical system (MEMS) microsensors can potentially give users unprecedented access to real-time situational information.

While sensor data are local to each node, the information content to be extracted from the network can be global, which must be obtained through collaboration among nodes. Let us consider a scenario of tracking chemical plumes using ad hoc just-in-time deployment of sensor nets:

The Valley Authority just declared a region-wide emergency: A large-scale hazardous chemical gas leak occurred ten minutes ago near the town of XYZ. The National Guard has been activated to evacuate nearby towns, and to close roads and bridges. To get a real-time situational assessment of the extent and movement of the gas release and aid evacuation, a SWAT Team is called in. Three unmanned aerial vehicles (UAVs) are immediately launched from an open field 15 miles south of the accident site, each carrying 1000 wireless chemical sensing nodes (Fig. 1). Upon flying over the vicinity of the accident site, the sensor nodes are released. The nodes self-organize into an ad hoc network once they get to the ground and relay the tracking result back to a base station nearby: Where is the plume? How big is it? What is the shape? How fast is it moving?

In this example, each sensor only has limited information such as whether certain chemical elements exist at the sensing spot, whereas the global information such as the shape of the plume and its motion need to be determined collaboratively by many sensors. In addition, because of limited resources (e.g., battery power and communication bandwidth), such processing and communication must be carefully managed.

The ultimate way to reduce energy consumption of a sensor node is by turning it off. Modern wireless sensor hardware platforms usually have low-power sleeping modes, in which parts of the processor, sensors, and wireless communication circuits are shut down to preserve power. For

1536-1284/04/$20.00 © 2004 IEEE

example, in Berkeley/Crossbow MICA2 motes,[1] 1 s of sleeping can save enough energy to send more than 70 packets, or performing ~70,000 CPU instructions. Sensor nodes can be turned back on by timers or by receiving wakeup packets using, for example, carrier detect circuits. Thus, the art of the system design is to selectively put sensors to sleep without losing application performance. This has traditionally been dealt with by adjusting the sampling and communication rate of the sensor nodes. In this article we push this philosophy further. That is, we use application-specific physical constraints to select nodes to be activated.

In a large-scale dense sensor network, it is sometimes desirable to process the information collected by the sensors within the network rather than sending raw data to a central server [1]. There are primarily two reasons. First is that the information collected by a sensor network is highly redundant. Considering that typical physical phenomena only have limited ranges of impact, most sensor data from a network contain no information about the phenomena of interest. Sending them out is simply a waste of resources. Second, given our current technologies in wireless sensor node design, wireless communication is still limited in bandwidth and expensive in power consumption. So even for meaningful sensor data, it is desirable to summarize the sensor data locally before sending to the edge of the network.

The set of sensor nodes (or computing agents in general) that collaboratively process the data within the network is called a *collaboration group*. There are many ways to define collaboration groups (e.g., based on geographical locations or data of interests) [2]. For instance, let us consider a point signal source to be tracked at (x, y) and its radius of impact r (defined by a certain signal-to-noise ratio); then it is intuitive to define a collaboration group that consists of all nodes within the circle defined by (x, y) and r. Of course, since the true location of the signal source is unknown, one may have to estimate its location and maintain the group accordingly [3]. However, when the phenomenon is non-local, as in the case of the chemical plume, it is not obvious how to find the minimum set of sensors that contribute to tracking the phenomenon.

This article reviews the dual-space transformation in computational geometry and applies it to the tracking of non-local phenomena [4]. We consider a dense sensor network so that the edge of a non-local phenomenon, modeled as a shadow, can be piece-wise approximated by straight lines. We study edge detection and tracking problems for a 2D continuous shadow over the senor field. A dual-space transform maps non-local line segments into a set of points in an appropriately parameterized configuration space. We then show how motion constraints from the target shape and dynamics can be exploited to activate only those sensors relevant to the current configuration. This algorithm can serve as a building block in a scalable hierarchical architecture that overcomes the communication and computation limitations. In our experiment of a shadow tracking using 16 motes, we have observed that only 28 percent of the sensors on average are awake at any given time.

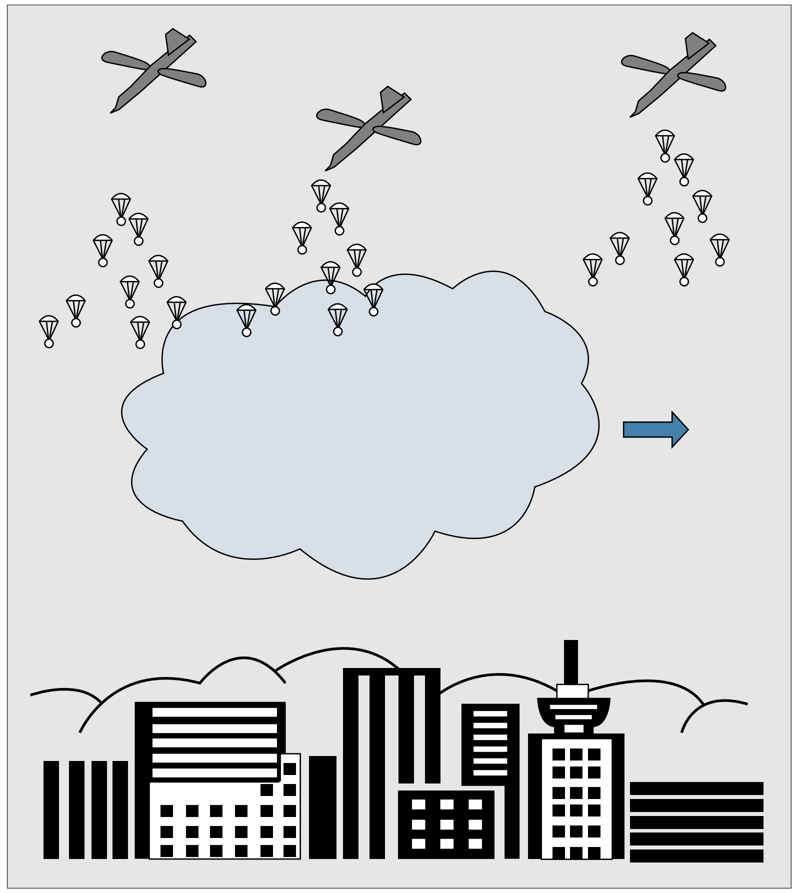


Figure 1. *Tracking chemical plumes using ad hoc distributed sensors.*

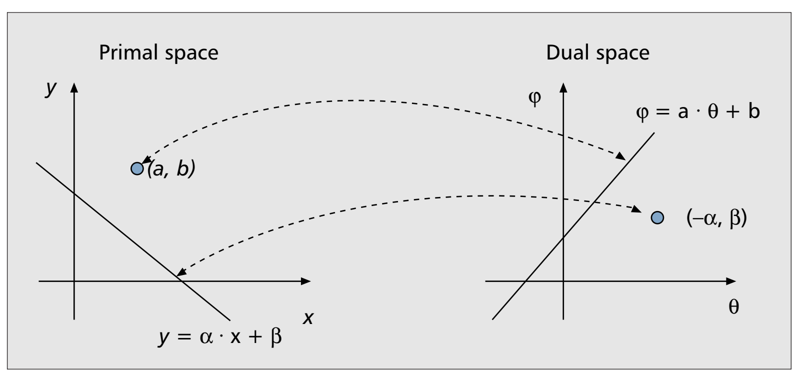


Figure 2. *The mapping between the primal space and the dual space.*

Dual-Space-Based Edge Detection

What does a dual-space representation buy us? The geometric duality described below allows us to map a seemingly non-local phenomenon (the position of the shadow edge) into a local attribute in the dual space. This allows the sensor nodes to be ordered according to how "close" they are relative to the frontier of the object motion and simplifies the sensor activation procedure. If the sensor activation algorithm were implemented in the primal space, without using the dual-space transformation, each sensor node would have to reason about its distance to the object edge relative to other sensor nodes and the motion of the object, a fairly complex geometric problem to solve. We consider a half-

[1] *Available from Crossbow Technology, Inc., http://www.xbow.com*

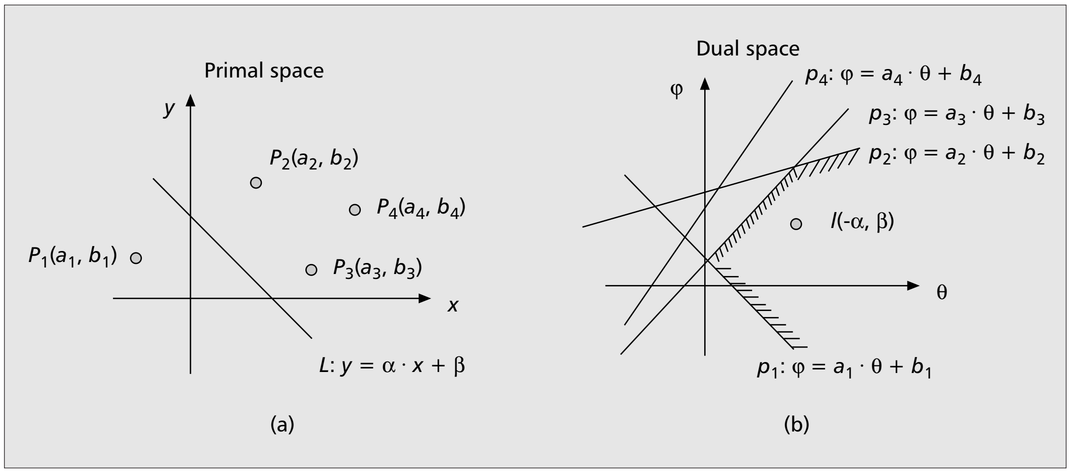


Figure 3. *a) A set of points and a line in the primal space; b) their representations in the dual space.*

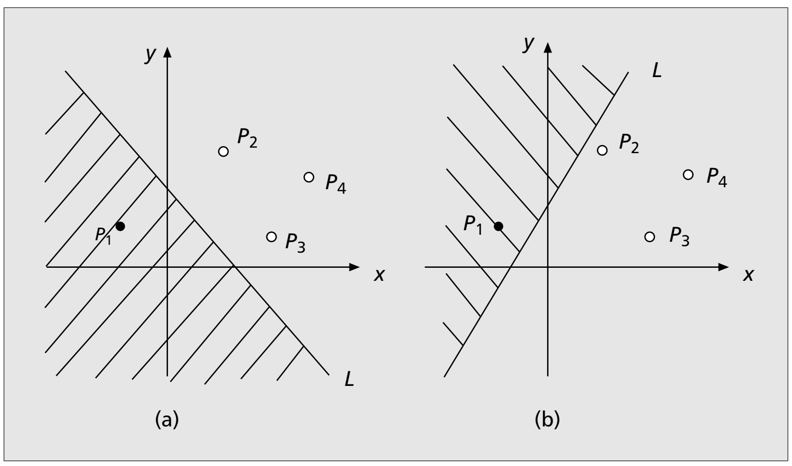


Figure 4. *Two different configurations that yield the same sensor reading.*

plane shadow in this section, and generalize it to other shapes later.

Dual-Space Transformation

Let us consider a line in a 2D space (called the *primal space*): $y = \alpha \cdot x + \beta$, which is uniquely defined by two parameters α and β. To represent this line through this pair of parameters, we can use the point $(-\alpha, \beta)$ in another 2D space (called the *dual space*).[2] Similarly, a point in the primal space (a, b) uniquely defines a line in the dual space: $\varphi = a \cdot \theta + b$. This 1-to-1 mapping, as shown in Fig. 2, is one form of a *dual-space transformation*[3] [5, 6].

A dual-space transform has several useful properties, which follow immediately from the definition:

A If, in the primal space, a point (a, b) is *on* a line $y = \alpha \cdot x + \beta$, in the dual space, the corresponding line $\varphi = a \cdot \theta + b$ goes through the corresponding point $(-\alpha, \beta)$, and vice versa.

B If, in the primal space, a point (a, b) is *above* a line $y = \alpha \cdot x + \beta$, i.e., $b > a \cdot \alpha + \beta$, in the dual space, the corresponding line $\varphi = a \cdot \theta + b$ is *above* the corresponding point $(-\alpha, \beta)$ (i.e., $\beta < -\alpha \cdot a + b$). Similar results hold for the below relation.

C If, in the primal space, a line $y = \alpha \cdot x + \beta$ performs a continuous motion, including rotation and translation, the corresponding point $(-\alpha, \beta)$ performs a continuous motion in the dual space.

For example, consider a set of points $\{\mathbf{P}_1, \ldots, \mathbf{P}_4\}$ and one line **L** in the primal space, as shown in Fig. 3a, whose corresponding dual-space representations, $\{p_1, \ldots, p_4\}$ and l, are shown in Fig. 3b.

In Fig. 3b, the lines $\{p_1, \ldots, p_4\}$ define a *line arrangement* that partitions the dual space into a set of convex polygons, called *cells* [5, 6]. The boundaries of these cells are line segments lying on the lines $\{p_1, \ldots, p_4\}$. Obviously, some cells are completely bounded, while others extend to infinity. The dual of a primal line $\boldsymbol{L}$ is a point l that must be contained in one of the cells (in this example, the shaded cell in Fig. 3b), unless it is on a cell boundary. By abusing notations, let us use $l < p$ to denote that point l is below line p in the dual space; then the shaded cell in Fig. 3b contains all points l satisfying $l > p_1$, $l < p_2$, and $l < p_3$.

When line $\boldsymbol{L}$ in the primal space moves, l moves in the dual space accordingly. As long as $\boldsymbol{L}$ does not rotate across the vertical direction or intersect any point in the primal space, l stays in the cell defined by the above set of constraints in the dual space. Furthermore, in the dual space, l can enter other cells only if it crosses one of the cell boundaries, including, conceptually, a boundary at infinity. In particular, as shown in Fig. 3b, l cannot intersect p_4 before it crosses one of the current cell boundaries. This observation is the key to our sensor selection scheme: if $\{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_4\}$ are the positions of four sensors and $\boldsymbol{L}$ is the boundary of a half-plane shadow, $\boldsymbol{P}_4$ can be safely turned off as long as none of $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, and $\boldsymbol{P}_3$ senses a transition.

Shadow Edge Estimation and Sensor Selection

For a half-plane shadow, by using the dual space transformation, we can estimate its edge by solving the set of constraints imposed by the particular sensor readings. Using that information, we can further determine the group of sensors at the "frontier" (i.e., the ones that may detect a

[2] *We use $-\alpha$ instead of α in the dual space so that some properties are easy to derive later on.*

[3] *It is also called a* Hough transformation *in some of the literature.*

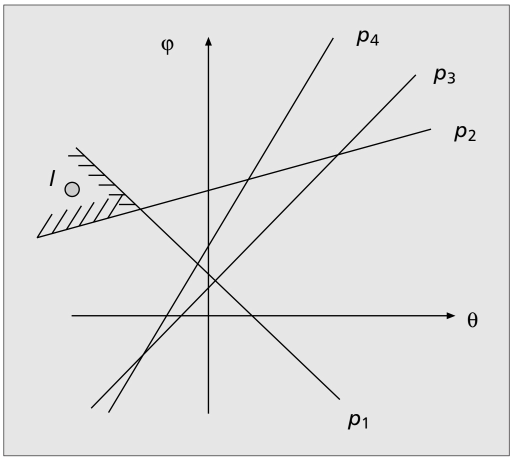
transition next when the shadow moves). For ease of discussion, we use light sensors as a metaphor for the sensing model. Obviously, the mechanism applies to any sensing models that give binary readings through quantization. Let **0** represent a *dark* reading at a sensor, and **1** represent a *light* reading. Then at any time, the sensor field gives a vector of readings consisting of **0**s and **1**s. The goal is to identify the set of sensors that bounds the shadow edge, and thus estimate the shadow location and turn off the nodes that are irrelevant at this time.

Using the dual space transform, each sensor defines a line in the dual space, and the edge of the shadow is a point. So the problem is converted to determining the cells that are consistent with current sensor readings; these are the cells that may contain the dual of the shadow edge. Note that the constraints in the dual space are in the form of *above* and *below* relations. The same vector of sensor reading may yield two possible answers for the location of the shadow: the shadow is above its edge or it is below its edge. For example, the two shadow locations shown in Fig. 4a and 4b yield the same sensor readings, [**0**, **1**, **1**, **1**], on {P_1, …, P_4}.

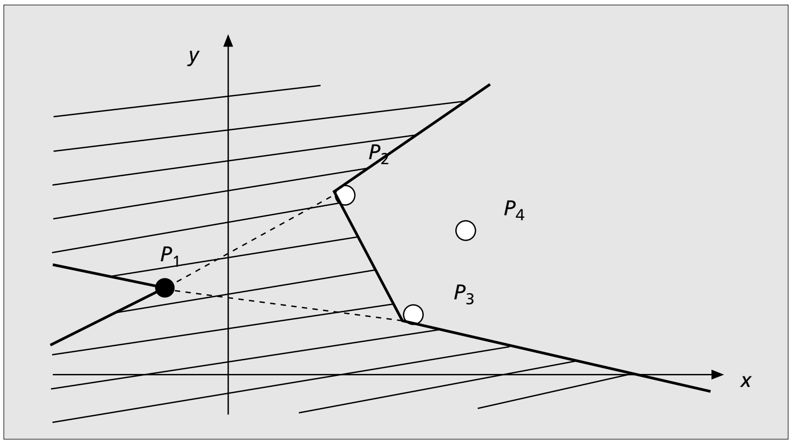
Moreover, in the dual space, situations in Fig. 4a and 4b have different representations. The representation of a is exactly the same as in Fig. 3b, while the representation of b is illustrated in Fig. 5.

The cells that are consistent with the set of sensor readings can be computed via linear programming over the results of a topological sweep. A topological sweep algorithm computes the segments created by the intersections of the set of lines, and their relative locations in terms of adjacency and direction. Using that arrangement, one may first assume that the shadow is above the line and use a linear programming algorithm to find the cell that is consistent with the set of sensor readings. A similar computation can be performed under a below assumption. Sometimes only one of the assumptions yields a feasible answer; sometimes both of them do. The details of the topological sweep and cell finding algorithms are out of the scope of this article. Interesting readers are referred to [4, 7, 8]. But it is worth noticing that these are all centralized algorithms that require location knowledge of all points.

Once we find the cells that satisfy the sensor reading constraints, the corners of the cells, which map to several lines in the primal space, we can determine the "extreme" positions where the edge of the shadow could be. Each pair of lines intersecting at a corner point, together with the corresponding constraints on the lines, give a wedge in the primal space. For example, in Fig. 3b the corner of the intersection of p_1 and p_3 together with the relations that the cell is above p_1 and below p_3 define a wedge that contains all lines above P_1 and below P_3 in the primal space. Similarly, the intersection of p_2 and p_3 and the fact that the cell is below p_2 and below p_3 give a wedge that contains all lines below P_2 and P_3. For each cell, the intersection of these wedges is the estimate of the shadow edge. That is, the edge of the shadow must be within that wedge under a certain assumption (e.g., dark means below). Of course, if there are two consistent cells in the dual space, the union of each shadow edge estimate gives the overall answer. For example, mapping back the cells in Fig. 3b and Fig. 5, we get the wedge shown in Fig. 6. In general, the size of the cells in the dual space dictates the "freedom" of the edge in the primal space. Thus, the smaller the cells, the more accurate estimation one can get.



■ **Figure 5**. *The dual space representation for the situation shown in Fig. 4b.*



■ **Figure 6**. *An estimation of the location of the shadow edge, subject to the resolution of the sensor field.*

Furthermore, at any time, the dual of the edge of a shadow can possibly be in at most two cells. The sensors corresponding to the boundaries of these cells are the sensors at the frontier, in the sense that no matter how the shadow moves, it must cross one (or more) of these sensors before crossing any other sensors. So only those sensors corresponding to the lines bounding the cell(s) need be kept active, thereby bringing the energy savings. It can be shown that in any 2D line arrangement the expected number of lines bounding an "average" cell is at most four [5] (in the model where all cells are equally likely), independent of the overall number of sensors present. Thus, we can expect the number of sensors that need to be active at any moment to be very small. In a dense sensor field this may lead to substantial energy savings over time without losing the tracking quality.

The ideas of tiered architectures and virtual grids have been shown helpful for scalable network services such as ad hoc routing and data dissemination. Here we apply similar ideas to sensor management.

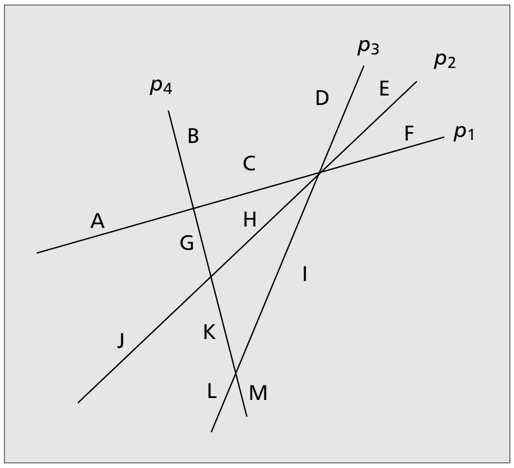


Figure 7. *A line arrangement of four lines.*

Distributed Sensor Management

The method described above for finding the cells in the dual space is static, and could be applied without knowing the motion history of the shadow. If we take advantage of the fact that the motion of a shadow is continuous, so the dual of its edge can only move from one cell to an *adjacent* cell, the computationally intensive linear programming part does not have to be performed after the system is properly initialized. For example, if the cell {C, G, H} in Fig. 7 contained the dual of the shadow edge, and sensor p_2 just flipped its reading, it is clear that {H, K, I} is the new cell. If the line arrangements are precomputed and stored in the sensor nodes, finding the new cell is simply a table lookup.

This process is so simple that it is amendable for a distributed implementation on tiny sensors with very limited memory and processing power. After giving each cell a unique ID, each sensor node only need remember those dual space cells that are incident on the line representing it. These cells correspond to the concept of the *zone* [5] of a line in a line arrangement, and it is known that the storage required is only linear in the total number of lines (nodes). The network can be initialized by having all sensors agree on the same cell ID. The sensor nodes that know nothing about that cell can go into sleep mode. When one of the sensors on guard notices a flip of sensor reading, it wakes up sensor nodes that are new in the new cell, and broadcasts to all awakened sensors the new cell ID. The sensor nodes forming the old cell but no longer in the new cell can put themselves into sleep mode.

A Laboratory Experiment

We have built an experiment [4] to validate the shadow tracking algorithms and demonstrate the benefits of sensor management using a network of Rene motes with light sensors [9]. The Rene motes are an older generation of Berkeley motes that have an 8-bit Atmel AT90LS8535 processor running at 4 MHz, 8 kbytes of flash memory, and 512 bytes of SRAM. One of the analog-to-digital (A/D) channels is connected to a light sensor. The sensor's sampling rate is programmed to be 8 Hz. RF communications between motes are carried on the 916 MHz band. Data are communicated at up to 10 kb/s.

The experiment is performed on a vertical 6 ft × 6 ft board to allow an overhead viewgraph projector to illuminate the entire platform. Sixteen motes are mounted on the board at randomized but known locations. Figure 8a shows the photograph of the board and the motes, which are numbered from 1 to 16.

Figures 8a and 8b show a scenario where the shadow covers mote 16. Five of the motes are elected as the frontier of the shadow tracking, which is obtained from the boundaries of the cell in the dual space (shown in Fig. 8c). In Fig. 8b, the lines connecting the frontier motes depict the extreme position and orientation of the edge of the shadow. In other words, the shadow's edge must lie within the bounds of all five lines.

In this particular setup, the 16 lines create a total of 102 cells, which cover all possible positions of the shadow edge. The number of boundaries for each cell indicates the number of motes that need to be activated when the dual of the shadow edge falls in that cell. In almost all cases (> 97 percent), only 3 to 5 out of the 16 nodes need to be active at the same time. In other words, less than 30 percent of the motes are active at any time on average. The rest can be put into sleep to preserve power.

A Two-Tier Architecture

There are still two potential pitfalls for the above sensor management scheme:

- Obtaining the cell configuration is computationally expensive, and the algorithm is centralized.
- In a large deployment, sensor nodes that form a particular cell in the dual space may be very far away in the primal space.

There may not be a direct communication link to wake up a remote sensor node if all not-on-guard nodes are sleeping. To overcome these limitations, we present a two-tier sensor management architecture over a virtual grid. The ideas of tiered architectures and virtual grids have been shown helpful for scalable network services such as ad hoc routing [10, 11] and data dissemination [12]. Here we apply similar ideas to sensor management.

The two-tier hierarchical architecture consists of a large number of featherweight sensor motes and a small number of more powerful (in terms of computation) nodes serving as cluster heads.[4] In order to maximize the lifetime of the network, both motes and cluster heads may have a deep sleep mode that consumes almost no power. We assume that they can also be awakened wirelessly via carrier sensing. As shown in Fig. 9, the sensor field is virtually divided into a regular grid. Each grid square (whose shape may not be exactly square) has one cluster head (shown as a star) and a set of motes (shown as circles) deployed in an ad hoc fashion. Both cluster heads and motes have wireless communication capabilities. In addition to communicating with the motes in its square, a cluster head can also communicate with other cluster heads in adjacent squares to create a mesh network topology (depicted as the hashed lines in Fig. 9). The

[4] *In fact, with sufficient computational power, the sensing nodes and cluster heads can be of the same kind, and their roles can be dynamically assigned after deployment to improve robustness and share load.*

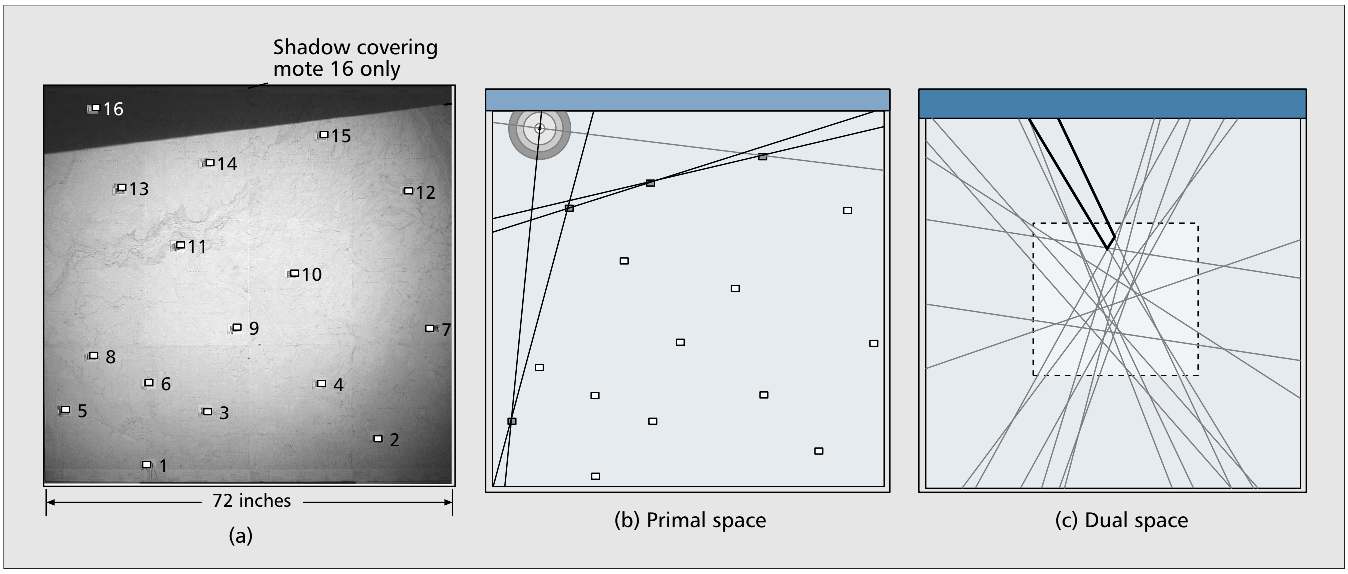


Figure 8. *a) A testbed contains 16 motes mounted on a board; a half-plane shadow is cast onto the board; b) the detection and estimation of the shadow edge; c) the dual-space representation of the corresponding cell. Motes 16, 15, 14, 13, and 5 are the current frontiers for detection.*

size of the grid squares is small enough that a broadcast from one mote can be heard by all the motes (and the cluster head) in that square, and it is big enough to minimize the total number of squares in the field. The cluster heads, which may not be equipped with any sensor, can be placed arbitrarily in the grid. We also assume that all motes are localized, presumably with the help from cluster heads.

Suppose that the boundary of the physical phenomenon to be detected and tracked (e.g., a chemical plume) is smooth such that it can be approximated as a straight line in each grid square. Then a dual-space-based detection, tracking, and sensor management scheme can be applied as follows.

Intracluster initialization: Once deployed, the motes send the cluster head their locations and current sensor readings. Using that information, the cluster head can perform a topological sweep algorithm and compute all cells in the dual space and the current cell(s). This configuration is then sent back to the motes in the form of a table.

Intercluster initialization: A grid square is called *covered* by the shadow if all its sensors have dark readings; it is called *uncovered* if all sensors have light readings; and it is called *partially covered* otherwise. The coverage properties are sent by the cluster heads to their direct neighbor cluster heads. For partially covered squares, their cluster heads also send the current estimate of the shadow edge. Upon receiving these messages, a cluster head decides whether to activate its cluster using the *grid activation* criteria:

- **A**: A partially covered square is always *active* in the sense that the frontier motes continuously sense the shadow.
- **B**: A covered or uncovered square is active only if at least one of its neighbors estimates that the edge of the shadow possibly intersects the grid boundary between them.

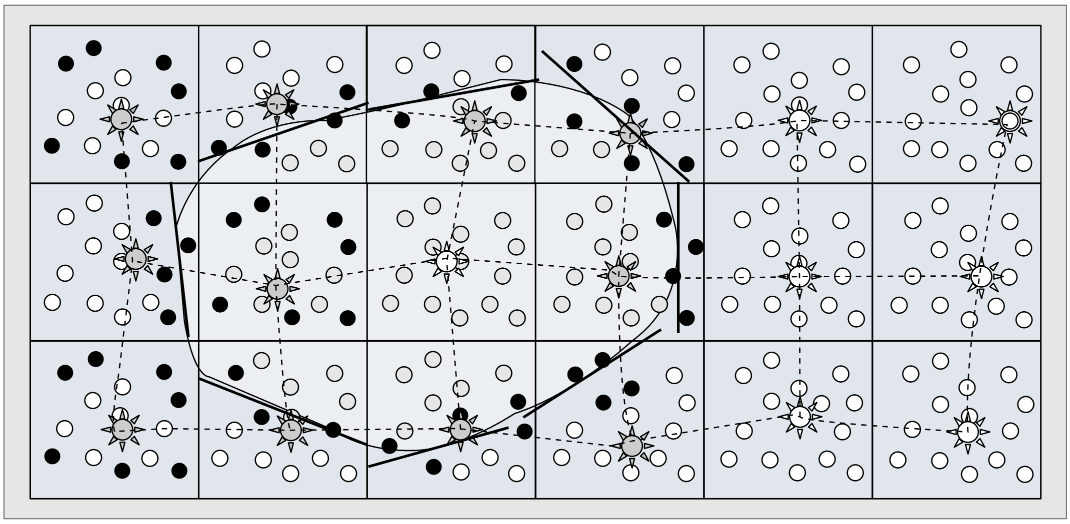


Figure 9. *Using a hierarchical architecture to track non-local phenomena.*

By converting non-local phenomena into localized representations and solving the problem in an appropriate configuration space, the sensor nodes at the frontier can be identified easily.

For an inactive square, all nodes are in sleep mode.

Tracking: An active square performs a tracking and sensor selection scheme using the algorithm described earlier. Motes that are not at the frontier of tracking go into sleep mode. Once the current dual-space cell in a square changes, its cluster head is awakened by the mote that senses the change. The cluster head tells all direct neighboring cluster heads about its new estimate of the shadow edge. The receiving cluster head then determines whether it needs to activate its own cluster by the grid activation criteria.

By using this scheme, only the sensors that are absolutely necessary in detecting and tracking the non-local phenomena are activated. Note that one of the assumptions we made in the system is that the edge of the shadow can be piecewise approximated as straight lines in each grid square. While the straight line approximation assumption is in general true, the extent of the lines may not be aligned with the boundaries of the grid squares. Future work is to dynamically create the clusters to adapt to complex shapes.

Conclusion

In large-scale dense sensor networks, the scalability requirements suggest we organize sensor nodes and computation into collaboration groups for in-network information processing. This article reviews the dual-space transformation principle and applies it to tracking the boundary of a non-local shadow. By converting non-local phenomena into localized representations and solving the problem in an appropriate configuration space, the sensor nodes at the frontier can be identified easily. Thus, other nodes can be safely switched to a power saving mode. To scale up this approach, we propose a hierarchical heterogeneous network with more powerful nodes performing dual-space transformation and long-range communication, and tiny motes performing fine-grained edge detection and tracking.

Acknowledgment

This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-00-C-0139 through the Sensor Information Technology Program. The authors would also like to thank Olaf A. Hall-Holt for helping on the topological sweep software, and Jim Reich and Juan Liu for inspiring discussions during this work.

References

[2] J. Liu *et al.*, "State-Centric Programming for Sensor-Actuator Network Systems," *IEEE Pervasive Comp.*, vol. 2, no. 4, Oct.–Dec., 2003, pp. 50–62.

[3] J. Liu *et al.*, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *2nd Int'l. Wksp. Info. Processing in Sensor Networks*, Palo Alto, CA, Apr. 2003, *LNCS* 2634, Springer-Verlag, pp. 113–28.

[4] J. Liu *et al.*, "A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks," *1st ACM Int'l. Wksp. Wireless Sensor Networks and Apps.*, Atlanta, GA, Sept. 28, 2002, pp. 131–39.

[5] M. de Berg *et al.*, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, 1997.

[6] J. O'Rourke, *Computational Geometry in C*, 2nd ed., Cambridge Univ. Press, 1998.

[7] H. Edelsbrunner and L. J. Guibas, "Topologically Sweeping an Arrangement," *J. Comp. Sys. Sci.*, vol. 38, 1989, pp. 165–94.

[8] E. Rafalin, D. Souvaine, and Ileana Streinu, "Topological Sweep in Degenerate Cases," *Proc. 4th Wksp. Algorithm Eng. and Experiments)*, San Francisco, CA, Jan. 2002.

[9] J. Hill *et al.*, "System Architecture Directions for Network Sensors," *Proc. 9th Int'l. Conf. Architectural Support for Prog. Languages and Op. Sys.*, Cambridge, MA, Nov. 2000.

[10] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. MobiCom 2001*, Rome, Italy, pp. 70–84.

[11] P. Krishna *et al.*, "A Cluster-based Approach for Routing in Dynamic Networks," *ACM SIGCOMM Comp. Commun. Rev.*, Apr. 1997, pp. 49–65.

[12] F. Ye *et al.*, "A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks," *MobiCom 2002*, Atlanta, GA, pp. 146–59.

[1] F. Zhao *et al.*, "Collaborative Signal and Information Processing: An Information Directed Approach," *Proc. IEEE*, vol. 91, no. 8, Aug. 2003, pp. 1199–1209.

Biographies

JIE LIU (liuj@microsoft.com) is a researcher at Microsoft Research. He received his B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, and his Ph.D. degree in electrical engineering and computer sSciences from the University of California at Berkeley in 2001. From 2001 to 2004, he was a research scientist at PARC. His research interests are modeling, simulation, and design of embedded systems, including system and software architectures, programming models, and synthesis tools.

FENG ZHAO (zhao@microsoft.com) is a senior researcher at Microsoft, where he manages the Networked Embedded Computing Group. He received his Ph.D. in electrical engineering and computer science from MIT, and has taught at Stanford University and Ohio State University. He was a principal scientist at Xerox PARC and directed PARC's sensor network research effort. He serves as the founding Editor-In-Chief of *ACM Transactions on Sensor Networks*, and has authored or co-authored more than 100 technical papers and books, including a recent book on wireless sensor networks published by Morgan Kaufmann. He has received a number of awards including NSF and ONR Young Investigator Awards and a Sloan Research Fellowship, and his work has been featured in news media such as *BBC World News*, *BusinessWeek*, and *Technology Review*.

PATRICK CHEUNG (pcheung@parc.com) is currently a research scientist in the Embedded Collaborative Computing Area of the Systems and Practices Laboratory at Palo Alto Research Center (PARC). He received his B.S. in electrical engineering from the University of Wisconsin at Madison, his M.S. in electrical engineering and control systems from the University of California at Berkeley, and his Ph.D. in controlling MEMS actuators from the M. E. Department of UC Berkeley in 1995. He holds 13 patents.

LEONIDAS GUIBAS'S biography was not available at the time of publication.